

A model-based approach to the preliminary design of a space tug aimed at early requirement's verification

*Original*

A model-based approach to the preliminary design of a space tug aimed at early requirement's verification / Cresto Aleina, Sara; Ferretto, Davide; Stesina, Fabrizio; Viola, Nicole. - (2016). (Intervento presentato al convegno 67th International Astronautical Congress, IAC 2016 tenutosi a Guadalajara (Mexico) nel 2016).

*Availability:*

This version is available at: 11583/2690023 since: 2017-11-23T12:06:33Z

*Publisher:*

International Astronautical Federation, IAF

*Published*

DOI:

*Terms of use:*

openAccess

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

IAC-16-D2.IP.11 (34331)

## A Model-Based Approach To The Preliminary Design Of A Space Tug Aimed At Early Requirement's Verification

Sara Cresto Aleina<sup>a\*</sup>, Davide Ferretto<sup>b</sup>, Fabrizio Stesina<sup>c</sup>, Nicole Viola<sup>d</sup>

<sup>a</sup> Department of Mechanical and Aerospace Engineering (DIMEAS), Politecnico di Torino, Corso Duca degli Abruzzi 24, Turin, Italy, 10129, [sara.cresto@polito.it](mailto:sara.cresto@polito.it)

<sup>b</sup> DIMEAS, Politecnico di Torino, Corso Duca degli Abruzzi 24, Turin, Italy, 10129, [davide.ferretto@polito.it](mailto:davide.ferretto@polito.it)

<sup>c</sup> DIMEAS, Politecnico di Torino, Corso Duca degli Abruzzi 24, Turin, Italy, 10129, [fabrizio.stesina@polito.it](mailto:fabrizio.stesina@polito.it)

<sup>d</sup> DIMEAS, Politecnico di Torino, Corso Duca degli Abruzzi 24, Turin, Italy, 10129, [nicole.viola@polito.it](mailto:nicole.viola@polito.it)

\* Corresponding Author

### Abstract

The paper deals with the design of a space tug involved in on-orbit satellite servicing missions through a Model Based approach. The space tug reference mission is defined in STRONG (Systems Technology and Research National Global Operations) program, inserted in space exploration and access to space frame supported by Italian Ministry of Research and University (MIUR). The space tug is a spacecraft able to transfer payloads from Low Earth Orbit (LEO) to higher operational orbits, thus allowing the reduction of subsystems complexity of the target spacecraft and a considerable optimization of its payload/platform ratio. Recently, space agencies are showing an increasing interest in space tug systems concept for the large range of future applications. After defining the mission architecture and Concept of Operation (ConOps), the work focuses on the application of a tool based on the integration of Model Based System Engineering (MBSE) elements in order to achieve an effective classification, traceability and verifiability of requirements among the various phases of the design process, combining the main features of specific tools and software, such as portability and flexibility, and the advantages of Model Based approach. In fact, the tool is aimed at guaranteeing an optimized data exchange among environments conceived for requirements management, design and simulation, allowing a coherent re-use of the information collected through specific analysis for others focused on different topics. The overall approach is based on the capabilities of the software, such as DOORS, Rhapsody, Capella, Matlab/Simulink, to maintain traceability of requirements during the handoff of the models, supporting requirements verification and allowing the realization of the multi-V approach. Indeed, it is demonstrated how this accurate management simplifies the planning and execution of the verification activities, because the requirements verification can be performed through in the loop simulation and test in any phase of the product life cycle. The paper shows the capabilities of integrated tools chain applied to the case study. The detail of the requirements of the space tug is provided highlighting how they derive from the mission scenario, the mission architecture, the ConOps and the functional analysis. Moreover, the recursive process of requirements definition and refining is properly managed, demonstrating how the proposed sequence of tools can help the verifications phases, saving time and money.

**Keywords:** Space Tug, Preliminary Design, Requirement Verification.

### Acronyms/Abbreviations

AD	Activity Diagram	MCC	Mission Control Center
AOCS	Attitude and Orbit Control Sub-system	MIUR	Italian Ministry of Research and University
BDD	Block Definition Diagram	MSC	Mission Support Center
ConOps	Concept of Operations	OBDH	On-Board Data Handling
DRM	Design Reference Mission	P/L	Payload
EPS	Electrical Power Sub-system	PRIDE	Programme for Reusable In-orbit Demonstrator for Europe
FFBD	Functional Flow Block Diagram	RD	Requirements Diagram
FMI	Functional Mock-up Interface	SAPERE	Space Advanced Project Excellence in Research and Enterprise
GEO	Geostationary Earth Orbit	SD	Sequence Diagram
IBD	Internal Block Diagram	SMD	State Machine Diagram
IXV	Intermediate eXperimental Vehicle	STRONG	Systems Technology and Research National Global Operations
LEO	Low Earth Orbit	SysML	System Modelling Language
LLO	Low Lunar Orbit		
MBSE	Model Based System Engineering		

TCS	Thermal Control Sub-system
TT&C	Telemetry Tracking and Control Sub-systems
UCD	Use Case Diagram
UML	Unified Modelling Language
VEGA	Vettore Europeo di Generazione Avanzata

## 1. Introduction

In past years, technological development and political environment have played a crucial role in space exploration history. However, while historically access to space was mainly a government prerogative, today the situation sees different figures. Indeed, there are many private companies in the world engaged in various space related fields: for example, there are companies that offer commercial launch services through private vectors (such as SpaceX) or companies which offer space tourism opportunities (such as Space Adventures). In addition, there are also many companies involved in the design of new aerospace devices and systems, as happens, for example, in the huge part of the market related to telecommunications and navigation services. A demonstration of this increasing interest can be seen in the amount of total financial resources applied to the space sector in recent years: the overall amount of financial resources related to the global space sector in 2014 is of about 330 billion dollars [1] (see Fig. 1). In addition, this value has seen also a significant growing trend from the previous year (2013): the overall expenditure in the space sector in the year 2014 is higher than the previous one of about 9% [1].

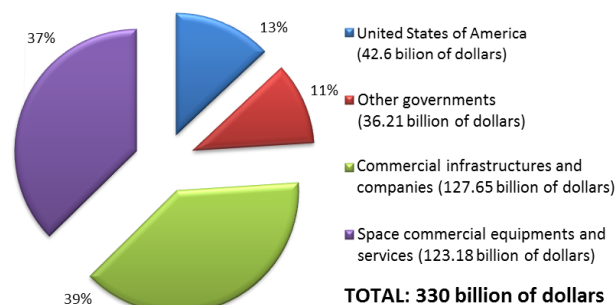


Fig. 1. Space sector global available financial resources distribution in 2014 [1].

Due to this increasing interest, the space is becoming even more crowded and the necessity to find new technological solutions able to reduce the number of orbital systems is even more concerning. For this reason, in recent years, international roadmaps draw the attention to a new space system concept: the space tug.

The Space Tug is a particular type of space vehicle designed as a reusable on-orbit spacecraft

applied for satellite servicing, developed to be adaptable to specific critical situations. An important application of this kind of system is the transfer of an on-orbit space system from LEO to higher operational orbits. The use of this kind of system in orbital transfer manoeuvres allows significant simplifications in satellite design, especially considering the propulsion system, with a consequent mass and volume reduction of the satellite. In addition, small launchers can be optimized to reach LEO, increasing the mass available for the payload that is no more supposed to reach the operative orbit through the help of dedicated on-board systems or through launcher stages. The only complication is in the need on the satellite to deliver of a dedicated docking system but solutions can be defined: for example, a standard and modular docking platform can be foreseen for the satellite to simplify its design. Therefore, this is not the only application of a Space Tug able to reduce the use of dedicated systems deployed on-orbit for a single servicing activity (see Fig. 2). For example, a Space Tug is an important building block in missions related to space exploration: issues regarding the assembly of large spacecraft can be solved using this system [2, 3]. Also Small Satellites can benefit from this kind of system: indeed, it is possible to consider the use of small launcher combined to a Space Tug to deliver out of the Earth sphere of influence Small Satellites applied, for example, in interplanetary missions, so designed to operate in orbit not easily reachable by small launchers [4]. There are even studies that suggest the use of tugs for the removal of asteroids if there is a risk of impact with Earth [5] as well as the debris removal on commercial orbit.

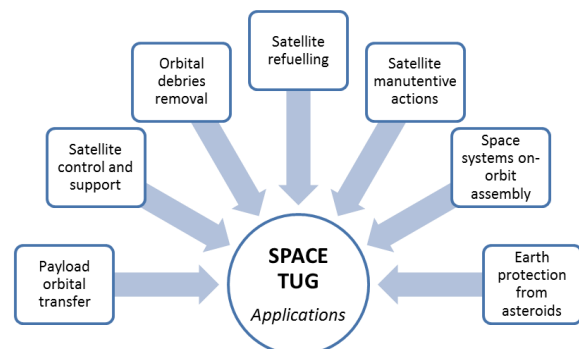


Fig. 2. Generic Space Tug applications examples.

One of the very first studies related to a Space Tug system design has been performed by NASA in the 70s. However, its production was not followed, mainly because the technologies available in those years were not mature to reach the performances required and because the design of the Space Shuttle was a priority [6].

Another example of study related to the design of a Space Tug is the SHERPA system. This Space Tug is a Spaceflight Inc. proposal for an orbital tug to be combined with SpaceX's Falcon 9 launcher and it could transfer small and secondary payloads to their operative orbits [7]. This Space Tug consists of a ring structure hosting the payloads and of a VASIMR (Variable Specific Impulse Magnetoplasma Rocket), theoretically capable of carrying several tons of payloads from LEO to Low Lunar Orbit (LLO) in few months. Particularly, it will host 5 payloads of 300 kg each, in the current version, providing a  $\Delta V$  of 400 m/s or 3 payloads of 300 kg each, in the version under development, providing a  $\Delta V$  equal to 2200 m/s [7]. SHERPA tug fly is scheduled for 2017.

Another example of study is the one output of SAPERE project and specifically of its STRONG sub-project that is the frame in which this activity is performed. This project has the objectives both to improve the national space operability in terms of access to space and to increase the Italian industrial capability to manufacture a Space Tug. The first objective is reached by analysing new operational mission concepts able to optimize the interfaces with the most important global space assets (e.g. VEGA launcher), through the Italian national know-how. The second objective is reached through the identification and development of those functions, peculiar of the Space Tug, complementing those already investigated in Europe and focusing for example on the validation of the Mars enabling technologies or on Active Debris Removal solutions. In particular, the STRONG Space Tug is an unmanned system deploying electric propulsion designed with the additional possibility to retrieve on Earth significant payload samples by means of an operative reusable vehicle, such as for example an evolution of IXV (Intermediate eXperimental Vehicle), Space Rider (previously PRIDE, Programme for Reusable In-orbit Demonstrator for Europe). In order to better describe the design activity performed on this kind of system, a methodology for the conceptual design has been studied and optimized for this kind of system, before applying it to the STRONG case study [8]. Through this study the mission architecture and ConOps of the STRONG tug have been analysed and characterized. Unfortunately, the results obtained through this previous analysis are not supported by software related tool and, for this reason, they are not able to achieve an effective classification, traceability and verifiability of requirements among the various phases of the design process even if a process was proposed. The presented work focuses on the application of this methodology in a tool based on the integration of MBSE elements on the STRONG case study, solving the previous work shortcoming. In this

paper the advantages of Model Based approach will be combined with the main features of specific System Engineering tools and software (e.g. portability and flexibility) in a proposed tool. Examples of application of software in a Model-Based approach can be seen in literature [9]. Particularly, in Section 2 the design of the STRONG tug will be presented focusing on the proposed conceptual design methodology. After this, the proposed tool will be described (Section 3), before applying it to the STRONG space tug design (Section 4). The main outputs of this section are to show how the application of system engineering tool and software in a structured methodology can drive the design of an existing case study simplifying the verification process. Eventually, main conclusions are drawn (Section 5).

## 2. STRONG Space Tug design

### 2.1 Methodology and tools

The main purpose of the presented work is the application of a conceptual design methodology in a tool based on the integration of MBSE elements. This application is addressed at the resolution of an important limit in the methodology proposed in [8]: the results obtained through this previous analysis are not supported by software related tool, as explained in Section 1. In this paper a tool chain designed in order to show the main advantages of both classical System Engineering processes and a Model Based approach will be proposed. In particular, the proposed tool chain shall be designed to support the typical conceptual design process (Fig. 3) [10, 11]. In this process the main output to be obtained is the definition of the requirements through the identification of the activities that such a system has to perform to be compliant with stakeholders' needs, regulations and other imposed constraints as, for example, the operative environment. The requirements definition process is important, considering that requirements represent the basis of the whole system design. For this reason, their derivation has to be part of a rational and logical process, in order not to forget drivers or constraints in the design that could eventually lead to unsuccessful choices. Also for this reason, a requirements categorization is necessary: as a matter of facts, having all the requirements divided into categories can reduce possible repetitions and helping their verification. For example, the main category of top-level requirements, i.e. mission requirements, directly stem out from the mission statement and mission objectives and constraints, which can provide a description of the crucial issue of this paper study and of the major limitations in the systems design. In

addition, other top-level requirements, for example programmatic requirements or constraints, are imposed from the analysis of all the actors involved in this project (defined as Stakeholder [12]).

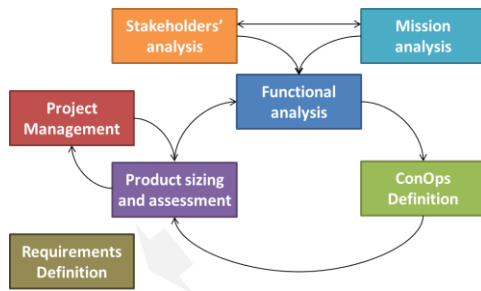


Fig. 3. Generic scheme of the proposed process.

The first activity to be performed, before writing down the requirements, concerns the definition of the main objectives of the project, that, as suggested in [12], come directly from the Mission Statement and stakeholders' analysis. In particular, primary Mission Objectives and Constraints are directly derived from the Mission Statement. On the other hand, Stakeholders' Analysis generates Secondary Mission Objectives and Constraints, through the definition of stakeholders' needs and expectations. Certainly, the stakeholders have first to be identified and categorized. As proposed in [13], the stakeholders can be classified as sponsors (i.e. people who establish mission statement and fix constraints on schedule and resources), operators (i.e. people in charge of controlling and maintaining the products), end-users (i.e. people that receive and use products and capabilities) and customers (i.e. users who pay fees to utilize a specific space mission's product).

At this point, the very next step is the identification of the main activities that the products have to perform in order to reach the objectives and guarantee the constraints. For this purpose, the typical Functional Analysis tools can be employed [10]. The main tool employed in the Functional Analysis is the Functional Tree, a tool able to define the basic functions (i.e. activities) that the system shall be able to perform. Secondly, the functions have to be mapped onto the elements able to perform them. This process can be also performed with tools, such as the Functions/Products Matrix: checked cells of the matrix are used to identify connections between functions and products, drawing the Product Tree. Then, an important aspect to be addressed is how they are organized and interfaced among each other: an example of approach to this feature is the Functional/Physical Block Diagram that is a graphical representation of the connections among all the products at each level of detail. In addition, this tool shows also the direction and the type of required

interfaces between products (e.g. data exchange or mechanical connection). In addition, also Functional Flow Block Diagrams (FFDB), which is a particular kind of tool that gives further information about timing and functional logical sequences, are adopted very often [10]. Being related to functions and not to products, this kind of tool shows what has to happen in the system without referring to physical solutions. On the contrary, a way to show the physical solutions that can be applied to solve the Mission Statement is the ConOps analysis. In particular, the definition of the ConOps should consider all the aspects of the mission to be performed, including integration, test, launch and disposal. Typical ConOps information are [10]: mission phases, modes of operation, mission timeline, Design Reference Mission (DRM) and/or operational scenarios, end-to-end communication strategy and/or command and data architecture, operational facilities, integrated logistic support and critical events.

Usually, in preliminary phases of the design process, it is common to have one or more operational scenarios and architectures, but only one is the optimal solution of the design. Trade-off analyses have to be performed in order to demonstrate which is the optimal solution, answering at the same time to the mission statement, the stakeholders' needs and the requirements.

It is important to remember that this process is iterative and recursive and has to be repeated from the highest level to lower levels until the desired level of detail, i.e. segment level, system level, sub-system level or device level. In each stage of the design process it is possible to define different types of requirements with different influences over the design. In addition, the previously exposed tools are examples that have to be readapted in a logical and rational process exploiting System Engineering software and a Model Based approach. It is true that the information achieved through these tools have still to be guaranteed in the new procedure and in the new comprehensive tool chain proposed.

## 2.2 *STRONG Space Tug*

The approach presented in the previous section has been applied to the STRONG space tug. Starting from the mission statement definition and the stakeholders' analysis, the main functions, products and operations related to the STRONG System of Systems has been studied with a particular focus on the Space Tug [8]. The main aim of this section is to summarize the main features of the STRONG Space Tug and of the STRONG scenario, while a complete analysis has been provided in [8] and [14].

The STRONG Space Tug is not the only element in the scenario. The elements of the overall STRONG

scenario include the VEGA launcher, every launch facility connected with the use of VEGA launcher, a payload (P/L) platform to be transferred, the Space Tug system, an orbital tank for on-orbit refuelling, the Space Rider vehicle, a Mission Control Center (MCC) and a Mission Support Center (MSC). These are the main elements of the mission architecture that, while interacting, will populate the mission scenario able to answer to stakeholders' needs and the mission statement. The ConOps includes the following phases: Space Tug deployment, Satellite platform deployment, Space Tug refuelling (Fig. 4). As a result of a trade-off analysis among different architectures, the refuelling configuration is constituted of an Orbital Tank to which the Space Tug has to dock for refuelling [15].

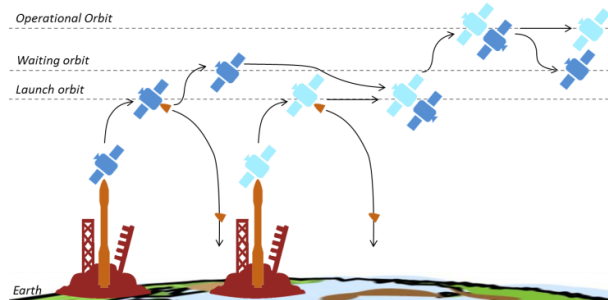


Fig. 4. Nominal electric space tug mission concept, the refuelling phase is not reported for simplicity.

In detail, considering this particular scenario as reference and the listed systems to be used, the first missions starts with the launch, through VEGA, of the space tug at a launch orbit (350 km of altitude and 5° of inclination). After being released in orbit, the tug is supposed to move autonomously in its waiting orbit (500 km of altitude and 5° of inclination) and remain there till the launch of a satellite platform. On the contrary, the tank is launched directly to the waiting orbit with a Soyuz launch. Consequently, VEGA launcher will bring P/Ls to be transferred, in the same launch orbit, while the tug has to reach the P/L. The maximum mass for a single P/L to be transferred is 1000 kg (from stakeholders' analysis). Once in the same orbit, the P/L is then docked to a Space Tug for the manoeuvres, thus allowing minimizing the propulsion on the platform and maximizing the P/L mass. Launch orbit and waiting orbit are supposed to be different. Once the tug has docked with the satellite platform at the launch orbit, the transfer towards the P/L final operational orbit begins. From stakeholders' analysis the maximum operative orbit to be reached is a Geostationary Earth Orbit (GEO) of 36000 km of altitude and 0° of inclination. After having released the P/L, the tug moves to the waiting orbit to perform the first refuelling. After that

refuelling operations have been completed, a second mission can start. In particular, 4 P/L transfers are supposed before a new Orbital Tank has to be provided.

In addition, considering this system architecture, the STRONG system will also give the opportunity to return on Earth significant P/L samples (Fig. 5). In this case, the Space Rider pre-operative reusable vehicle can be exploited in order to return the P/L (or some sensitive samples of it) from the waiting orbit to Earth after having transferred it on-board through a robotic arm. At the end of both cases, a refuelling phase is then required to extend the Space Tug reusability considering also that stakeholders' constraints impose to have a complete P/L transfer (or retrieval) in not more than a year.

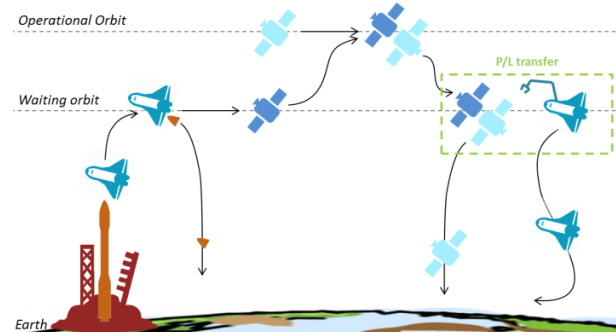


Fig. 5. Nominal electric space tug mission concept.

In both the presented cases, one of the main constraints in the Space Tug configuration is related to the compatibility with VEGA capabilities in terms of mass and volume (maximum diameter 2.6 m and maximum length 7.8 m [16]): this constraint has a significant influence on the choice and the design of the Space Tug sub-systems.

In particular, the Space Tug will be equipped with a certain number of sub-systems, including Propulsion Sub-system, Electrical Power Sub-system (EPS), Thermal Control Sub-system (TCS), Attitude and Orbit Control Sub-system (AOCS), On-Board Data Handling (OBDH) Sub-system, Telemetry Tracking and Control Sub-systems (TT&C), Structures Sub-system, Harness Sub-system [8]. The Propulsion sub-system includes the thrusters, the reaction control system, propellants tanks, all the interface and feeding devices needed to provide propellant to the thrusters and the active refuelling system to interface with the Orbital Tank. In particular, electric thrusters with a power of 9.6 kW will provide a constant thrust equal to 480 mN and an Isp of 2500 s. In addition, thrusters' power ratio is assumed to be of about 50 mN/kW. A very impacting sub-system is the EPS, since the tug is equipped with electric thrusters and, this system is in charge of



providing, storing and distributing power to the other sub-systems. EPS mainly includes solar arrays (with an area of 62 m<sup>2</sup>) and batteries (with a capacity of 9 kWh and a specific energy of 175 Wh/kg). Another enabling sub-system is the AOCS, aimed at stabilizing the system and orienting it in desired directions during the mission despite of external disturbance torques. The attitude control is also particularly critical for the rendezvous and docking manoeuvres required. Finally, another compelling sub-system is the structure one, that supports all the other sub-systems and includes the attachment interfaces with the launcher and the ground support equipment interfaces. Moreover, it includes the rendezvous and docking mechanism to dock with the P/L platform and with the tank. The final tug dry mass is of about 1361 kg (Table 1) while the other part of available mass on the VEGA fairing is considered to be filled with enough propellant to perform a complete P/L transfer.

Table 1. Space Tug mass breakdown.

Sub-system	Mass fraction	Margin	Mass
<i>Propulsion</i>	25%	10%	285 kg
<i>EPS</i>	27,34%	15%-20%	310 kg
<i>TCS</i>	4,50%	20%	51 kg
<i>AOCS</i>	8,38%	5%	95 kg
<i>OB DH</i>	2,38%	20%	27 kg
<i>TT&amp;C</i>	2,20%	15%	25 kg
<i>Structures</i>	25,84%	20%	293 kg
<i>Harness</i>	4,23%	20%	48 kg
<b>TOTAL</b>	-	20%	1361 kg

Further data about the mission and the required performances have been obtained through an iterative sizing and trade-off analysis, please refer to [8] for an explanation of these results.

### 3. Overview of the selected tools

In order to develop a Model-Based design process, starting from the methodology described in Section 2, a toolchain shall be selected to support the implementation of the different aspects and phases. Several tools and software for MBSE have been considered and their selection is described briefly in this section, with particular focus on their features and on their role within the overall process.

Requirements definition and management during the design process is fundamental. The development of the system of interest is, in fact, driven by the specification because its functional and physical characteristics are specified by requirements statements, and, on the other hand, the evolution of system architecture may produce new requirements, contributing to the update of the specification itself.

For these reasons, an easy accessible requirements database, shared among different platforms and tools, is necessary within the toolchain. The IBM Dynamic Object Oriented Requirements System (DOORS®) [17] was selected as main hub for the requirements, due to its wide application in MBSE and the capabilities of supporting connection with multiple design tools and software. DOORS® is a robust database organized in a predefined way, with projects, folders and modules, that are used to collect, classify and link requirements. Modules are the most important objects of the hierarchy since they contain data related to the items stored within the database. These objects represent the requirements themselves and some other elements of the specification, being the atomic parts of the structure of the tool, and can be classified as formal and link modules. Formal modules contain the list of requirements, eventually organized following the design phases, whilst link modules are used to map the links that are present among the requirements inside the specification, in order to specify the derivation structure of low level objects from high level ones. With this hierarchy, each requirement can be easily identified and traced during the whole design process, as it will be presented in Section 4, where the implementation of the traceability is shown. Particularly, the traceability which is internal to the specification is already guaranteed by the link modules, whilst the traceability with the system architecture (external) can be implemented through the deployment of the MBSE tool used for the design, which shall be interfaced with DOORS®. The choice of the software where the functional and operational design shall take place is mainly driven by the possibility of integration with the requirements database and with other software aimed at numerical analyses and simulation, as well as by the capability of supporting the methodology described in Section 2 in terms of tools used for the derivation of system characteristics. Taking that in mind, it has to be noted that different software are available on the market, so, a sort of trade-off shall be made to evaluate the best solution, following the aspects just mentioned. A quite interesting and open source tool is the Eclipse-based Capella® platform by PolarSys, which implements the Arcadia methodology [18]. The most important aspect that makes this tool interesting for the application of MBSE is that the design process is already provided out of the box. The Arcadia methodology is formalized within the tool and the user is responsible only to apply it to the case study, following the different steps. Another important point is that the typical tools used for the design are supported, making the implementation of the methodology described in Section 2 quite easy. However, some

important drawbacks are also present from the point of view of interoperability with other software. The connection with a requirements database is poor as well as the communication with numerical analyses. Moreover, the presence of the Arcadia process pushes the user to apply a specific method, making the strong point of the tool a weakness in case of general purpose use. Other Eclipse-based software are available for this kind of use, like Papyrus® [19], which is another open source platform supporting different modelling languages, as UML and SysML [20, 21], conceived to provide a wide range of opportunities. Unfortunately, the problems related to interoperability are still present and the considerable number of possibilities offered to the user makes the tool more complex than can be imagined at a first glance. The same applies to similar open source software, as Modelio® [22]. The idea that a general purpose environment with high interoperability capabilities is the best solution for the case study brought the attention on two commercial tools: Integrity Modeler®, by PTC, and Rhapsody® from IBM [23, 24]. The platform from PTC derives directly from the former Atego Artisan Studio. The Modeler® is a solution for MBSE with SysML and UML, fully integrated with the PTC Integrity® family, supporting interoperability with DOORS® and simulation tools like Mathworks Simulink®. Similar characteristics can be listed for the IBM Rational Rhapsody® product, which is a modelling environment for UML and SysML, supporting some architectural frameworks, that is part of the IBM Collaborative Lifecycle Management® platform. The main difference with its competitor consists in the possibility of choosing a proprietary IBM methodology, called Harmony® [25], for the design of the system of interest. On the other hand, it can be used as general purpose environment for modeling, not forcing the user to adopt vendor's methodology. Moreover, the advantage of being a solution developed by the same company allows a better integration with DOORS®, enabling some interesting features, as the coverage and impact analysis of requirements. Integration with simulation tools like Simulink® is again guaranteed even if a higher knowledge of the tool is required by the user if compared to the export facility of the Integrity Modeler®, which results easier to understand. However, considering the advantage of an ad-hoc built integration with requirements database and the wide range of possibilities offered for system modelling, Rhapsody® was chosen as MBSE environment for system design. As explained in Section 4, it will be used to characterize functional and operational aspects of the system in different phases, from stakeholders and mission analysis to the

lower levels of system definition (subsystems, components, devices etc.) using SysML. Moreover, it will be adopted to create the fundamental views and diagrams presented in Section 2, to trace and allocate requirements to functional and physical architecture, to establish the interfaces within the subsystems and components of the system in order to prepare the numerical simulation in terms of block diagrams and system breakdown. The two-ways link with the requirements database allows synchronizing and updating the specification either from DOORS® or from Rhapsody® itself, allowing an effective integration of requirements and system elements. Moreover, the possibility of preparing the data for further types of analyses, as simulation, allows creating a seamless oriented toolchain, reducing the time related to models set-up in separate environments. Particularly, the connection with Simulink® is available with a dedicated import/export facility even if the tool is also able to support the interoperability standard Functional Mock-up Interface (FMI) for model exchange [26, 27].

Some final remarks shall be expressed also for the simulation tools. Simulink® was chosen as main platform for this kind of application due to the wide range of possibilities in terms of uses and integrations, even if some open source tools, like OpenModelica® [28], and other commercial software, as AMESim® [29], were evaluated. The final choice was also driven by the availability of some in-house tools aimed at Model-Based verification that are currently supporting integration with Matlab/Simulink® [30] and which will be used in future works for the verification campaigns. Finally, the opportunity of integrate Simulink with both DOORS® and Rhapsody® allows to complete the design loop, from requirements up to the implementation of the simulation for the system design, through the definition of functional and operational aspects, back to requirements specification again.

#### **4. Implementation of the Model-Based design process**

As it was introduced in Section 2, the overall design process is deeply characterized by iteration and recursion, thus requiring for the implementation approach to be well structured in terms of phases and organized for what concerns toolchain integration. This approach is based on SysML modelling in Rhapsody® for what is related to Stakeholders and Mission analyses, Functional analysis and ConOps analysis, where the main diagrams are created to describe system functional, operational and physical architecture. Requirements analysis, which continues



during the whole process, is performed jointly in Rhapsody®, where requirements are defined, and in DOORS®, where they are stored, classified and ordered. Functional and ConOps analyses are also divided in sub-phases in order to maintain consistency with the depth of system characterization, notably STRONG level (top level), segment level, system level (where the Tug is defined), subsystem level and device level.

The Stakeholder and Mission analysis has been performed through a use case analysis, where the relations among stakeholders and mission objectives, both primary and secondary, have been represented in a Use Case Diagram (UCD) that stands as graphical representation of the mission statement. As it can be seen from Fig. 6, the use cases indicate the objectives that the stakeholders want to reach by using the system, whose borders are sketched by the boundary box in the center of the figure, whilst the stakeholder themselves are clearly shown outside of it.

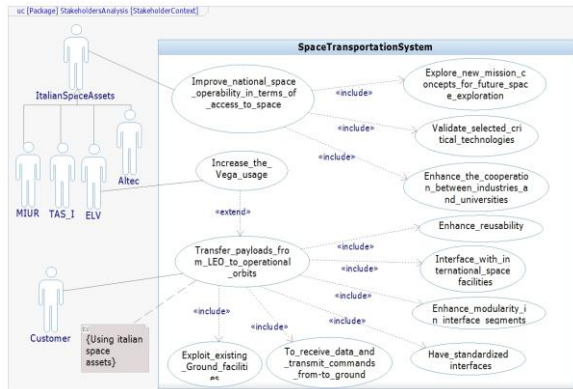


Fig. 6. Use Case Diagram for Stakeholders Analysis

SysML dependencies are used to state the relations between primary and secondary objectives, whilst generalization is used to specify the different Italian space assets. From these preliminary objectives, the first set of mission requirements can be derived through the Requirements Diagram (RD), where use cases are traced onto them. Requirements are then synchronized to DOORS® thanks to the proprietary Rhapsody® Gateway and listed in a dedicated formal module. This first set of elements and relations, which can appear quite simple, is fundamental for the following Functional and ConOps analyses since it provides the basis for the traceability links that will be populated and extended during the process.

Functional analysis is organized in different phases, as previously explained, and split in two main packages in order to look both at system functionalities and at the products onto which these functionalities will be allocated. The first step

concerns the creation of the functional tree, which is implemented through a Block Definition Diagram (BDD). This diagram represents the functional breakdown for the specific level of analysis, showing the hierarchy levels among the blocks. Fig. 7 represents the BDD for the breakdown at segment level.



Fig. 7. Block Definition Diagram for segment functional breakdown of STRONG system

As it can be seen, blocks are derived from the top level function in order to cover the required system functionalities also coherently with the previously defined mission objectives. Each block (which is actually a function) is described by an Activity Diagram (AD) where the relations among the functions are highlighted and the sequence of their execution is presented as happens for FFBD. These diagrams help a lot the derivation of low level functions and propose an important sketch for the further definition of ConOps architecture. Object nodes have been used to guarantee a coherent link between the functions and the element of the AD for traceability purposes.

Moreover, thanks to the defined functions, new functional requirements can be derived, transferred to DOORS® and linked to functional block to keep the traceability path unambiguous. Functional segment requirements are also linked to higher level requirements in Rhapsody® to highlight the derivation process, whilst it is possible to replicate this kind of relations in DOORS® thanks to a dedicated link module. The link module is a powerful tool to trace the relations among requirements directly within the database, exploiting the so-called internal traceability (Section 2). Different link modules have been defined by establishing proper link sets between the formal modules related to the several phases. This will allow browsing the derivation structure directly within the requirements database, from mission requirements to device requirements. Fig. 8 shows an

example of link module, established between mission and segment requirements.

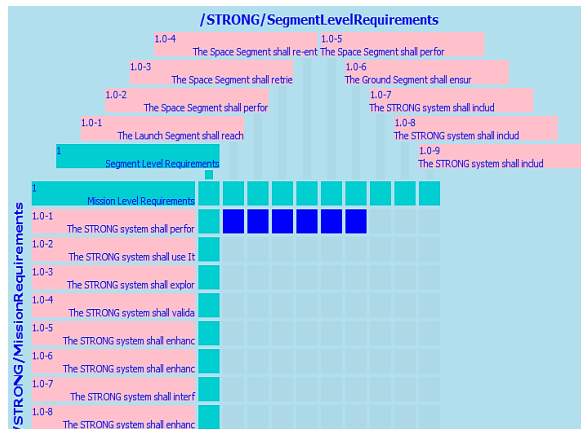


Fig. 8. Link module used to establish relations between Mission and Segment Requirements.

Similar views can be created also in Rhapsody® where dedicated matrix layouts can be configured in order to obtain a visual summary of requirements derivation and functions-requirements coverage. This process has been replicated for each phase building a considerably high number of relations among model elements and requirements, and constituting a solid multi-tools platform for traceability.

A similar approach has been adopted for the products architecture. Product tree was again represented as BDD and organized following the phases of the analysis. Fig. 9 shows the BDD concerning the segment product breakdown.

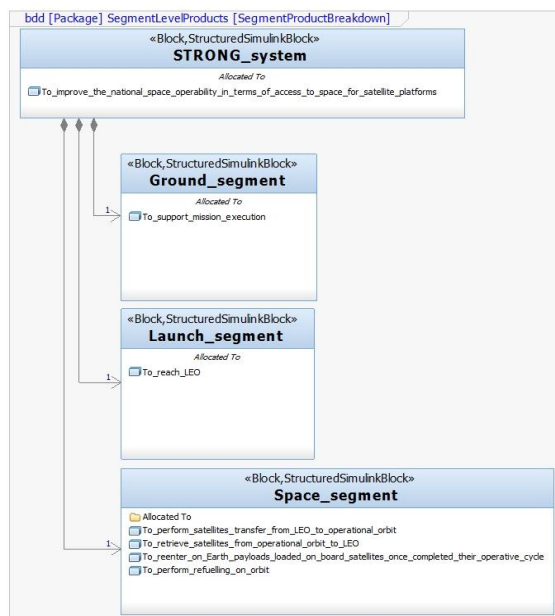


Fig. 9. Block Definition Diagram for segment product breakdown of STRONG system

As it can be seen, the blocks are organized following the same method used for functions. In Fig. 9 the product blocks contain the information about the functions that they are responsible to accomplish since functions have been allocated to them through proper dependency links. The specular information is present inside the functional blocks as shown in Fig. 7. Functions/products matrices can be created to summarize these mutual relations. Requirements are then connected to products through a stronger type of relation, the so-called satisfaction, meaning that the specific aspect stated by the requirement will be formally accomplished by the related product, even if a real verification is not yet present. This type of dependency concludes the path of the requirement, which started from the derivation, passed through the trace link onto the function and ends now onto a product. Another important aspect to be considered within product architecture definition is related to the formalization of the internal interfaces among the product themselves, that can be useful not only to sketch the topology of a specific layer of the system, but also to introduce other types of analysis, like simulation and verification campaigns early in the design process, and to raise the automation level related to data sharing among tools. For these reasons, Internal Block Diagrams (IBD) have been created to specify the internal structure of the blocks and to define the proper interfaces among them. Fig. 10 shows the IBD for the space segment block.

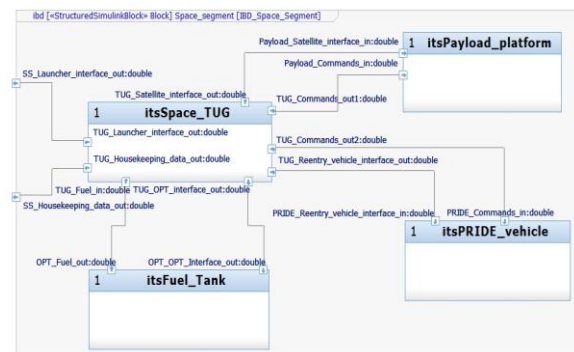


Fig. 10. Internal Block Diagram for the space segment

As it can be seen, the diagram includes a simple representation of the elements inside the space segment, with a preliminary definition of the internal interfaces. This structure can be eventually replicated in Simulink®, since the ports, the signals and the variables can be exported after a dedicated set up procedure. In general, IBD can be used as Model-Based version of Physical Block Diagram and they can be customized at user discretion for multiple purposes. As it was described for functions, the

analysis of products breakdown and architecture is replicated onto the different phases up to device level.

Together with Functional analysis, the ConOps analysis is realized following the same structure of design phases. However, it starts from a different assumption, taken as initial model element, which is constituted by a dedicated UCD used to define mission scenario at different level of depth. For this application, use cases represent the phases, sub-phases or particular operational situation where the system shall be able to work, whilst the external actors stand for the external instances that will have to interface with the STRONG system or will simply affect system behavior. Fig. 11 shows the UCD for segment level ConOps, where the links between STRONG mission and the other use cases are highlighted, together with the associations with the products at segment levels (which are inside the boundary box because they are part of STRONG system). An example of how external actors are involved is even provided.

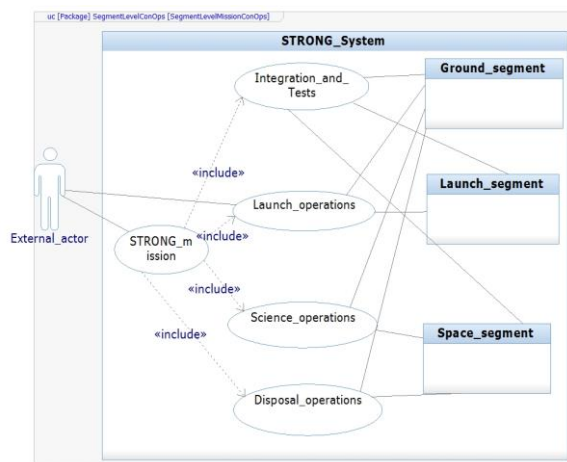


Fig. 11. Example of Use Case Diagram for segment level ConOps

Each use case can be then characterized through Sequence Diagrams (SD) and/or State Machine Diagrams (SMD) to specify sequences and modes of operations of the system, defining the so-called use case realization. The connection between ConOps and Functional analysis is based both on dedicated traceability links established between the use cases (mission phases) and functions, since those are literally used during the different phases, and by the presence of the products previously defined. ConOps analysis represents the most interesting field of the modelling activity, since the use of different behavior diagram of the SysML allows representing several aspects and views of the system in operations. A good way to summarize the overall process is looking at the web of traceability links that is

recorded thanks to the coverage analysis features of the Rhapsody Gateway®. Fig. 12 shows, as example, a view of the project where the relations of the function related to the capabilities of on orbit refueling with some mission objectives and segment requirements are expressed.

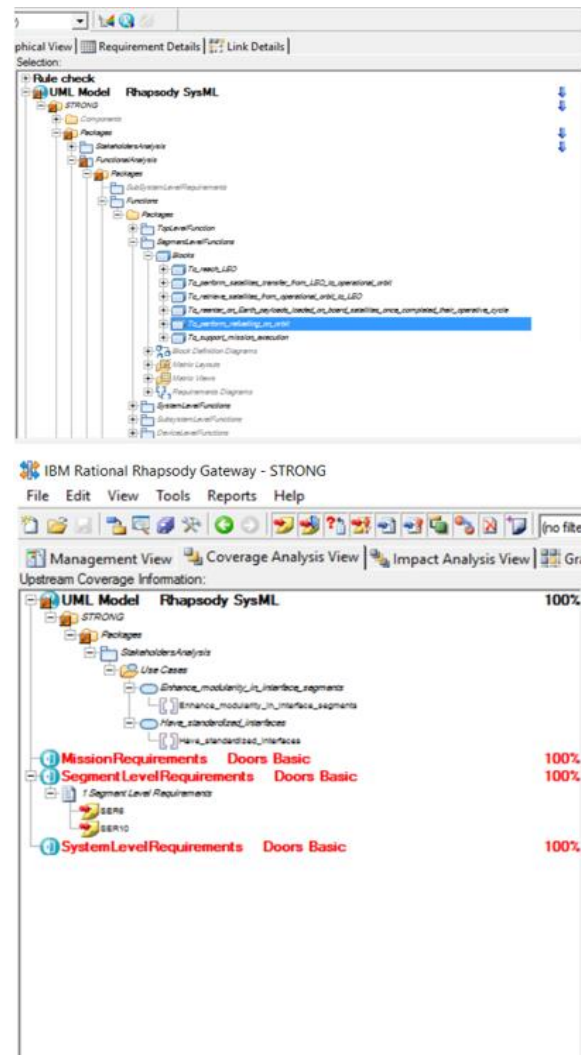


Fig. 12. Coverage analysis with Rhapsody Gateway

However, Rhapsody® allows navigating the whole set of relations of an element even without the dedicated analyzer embedded in the Gateway. Fig. 13 shows the properties tab of the same function considered in Fig. 12. As it can be seen, many data are present and notably: the diagrams where the function block appears (AD, IBD, BDD and RD), the dependency links respectively to mission objectives, to object nodes in FFBD, to requirements, to mission phases and the allocation to products.

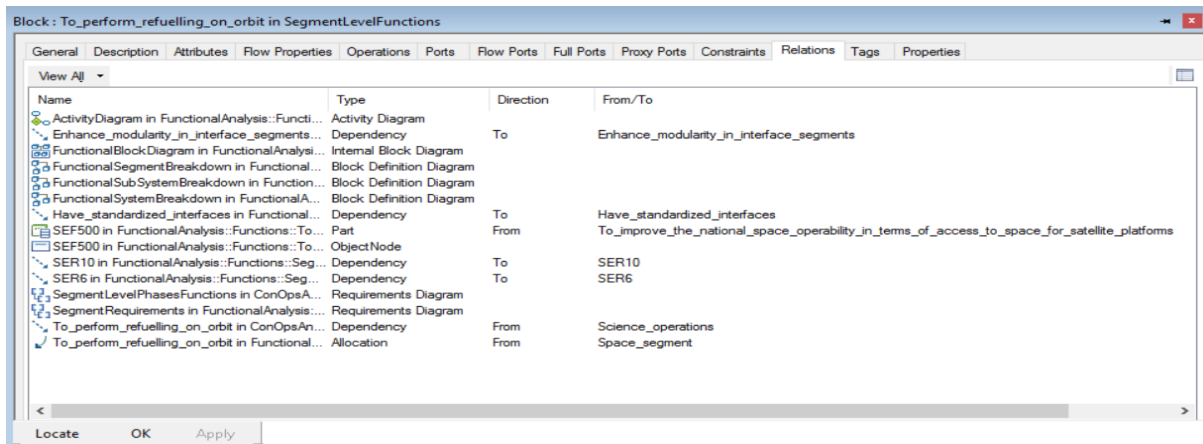


Fig. 13. Overview of the properties of a functional block

This huge amount of information is available for each model element, and it is updated live during the design process, enhancing considerably the quality of traceability and solving those problems related to data classification for Document-Based procedures.

## 5. Conclusions

This paper proposes a tool designed in order to merge and share the main advantages of both classical System Engineering processes and a Model Based approach, with the future purpose of simplify verification processes in late design phases. This tool is addresses at the resolution of an important limit in classical System Engineering processes: the results obtained through these processes are, indeed, not supported by software related tool and so are not able to achieve an effective classification, traceability and verifiability of requirements among the various phases of the design process. In particular, the proposed tool is based on the typical design process, in which the main outputs to be obtained are the requirements, defined through the identification of the activities that such a system has to perform to be compliant with stakeholders' needs, regulations and other imposed constraints. Considering all the tools and software for MBSE available on the market, the final toolchain has been implemented exploiting IBM's DOORS® and Rhapsody®.

The application of the proposed toolchain to a known case study (i.e. STRONG space tug) has preliminary demonstrated the possibility to simplify the application of classical System Engineering processes, increasing the classification and traceability of requirements among the design activity. In addition the use of this kind of toolchain will also increase the verifiability of requirements during and at the end of the design loop, allowing a reciprocal simplification of both the system design and the product realization processes through a continuous verification of the requirements.

Indeed, future developments of this work will focus on the definition of a methodology to exploit the proposed toolchain to simplify verification and validation processes among the various phases of the design process. Particularly, the demonstrated effective classification, traceability and verifiability of requirements will be exploited to complete the design loop, creating continuously during the design process inputs for verification and validation activities. Important for this phase is the high integration of the chosen tools with simulation environments such as Matlab® and Simulink® or ad hoc developed tool, which are easily configurable to be a dynamic link between the design and the verification phases.

## References

- [1] American Space Foundation, The Space Report 2015: The Authoritative Guide to Global Space Activity, available on [www.SpaceFoundation.org](http://www.SpaceFoundation.org), (accessed 26.08.2016).
- [2] M. A. Viscio, E. Gargioli, J. A. Hoffman, P. Maggiore, A. Messidoro, N. Viola, A methodology to support strategic decisions in future human space exploration: From scenario definition to building blocks assessment, *Acta Astronautica*, 91, pp. 198-217, <http://dx.doi.org/10.1016/j.actaastro.2013.06.015>, 2013.
- [3] M. A. Viscio, E. Gargioli, J. A. Hoffman, P. Maggiore, A. Messidoro, N. Viola, Future space exploration: From reference scenario definition to key technologies roadmaps, *Proceedings of the International Astronautical Congress, IAC*, 11, pp. 9131-9145, 2012.
- [4] F. Stesina, M. A. Viscio, N. Viola, S. Corpino, S. Fineschi, F. Fumentì, C. Circi, Interplanetary CubeSats system for space weather evaluations and technology demonstration, Elsevier Ltd *Acta Astronautica*, Volume 104, Issue 2,



- November 2014, Pages 516-525. ISSN: 0094-5765,  
<http://dx.doi.org/10.1016/j.actaastro.2014.06.005>, United Kingdom, 2014.
- [5] R. L. Schweickart, E. T. Lu, P. Hut, C. R. Chapman, The Asteroid Tugboat, Scientific American, November 2003
- [6] E. F. Harrison, E. B. Pritchard, Use of Space Tug to increase payload capability of Space Shuttle, Langley Research Center Hampton, Va. 23365 National Aeronautics and Space Administration, Washington, D.C., February 1971.
- [7] J. Andrews, Spaceflight Secondary Payload System and SHERPA Tug – A New business Model for Secondary and Hosted Payloads, SSC12-V-6, 26<sup>th</sup> Annual AIAA/USU Conference on Small Satellites, 2012.
- [8] S. Cresto Aleina, N. Viola, F. Stesina, M. Antonietta Viscio, S. Ferraris, Reusable space tug concept and mission, Acta Astronautica, Volume 128, November–December 2016, Pages 21-32, ISSN 0094-5765,  
<http://dx.doi.org/10.1016/j.actaastro.2016.07.003>.
- [9] B. Di Giandomenico, C. Pessa, E. Valfré, I. Viglietti, Alenia Aermacchi needs and experience on MBSE: the Crystal user scenario, 23rd Conference of the Italian Association of Aeronautics and Astronautics (AIDAA), Politecnico di Torino, 17-19 November, 2015.
- [10] NASA Systems Engineering Handbook, NASA/SP-2007-6105 Rev1, National Aeronautics and Space Administration, NASA Headquarters Washington, D.C. 20546, December 2007.
- [11] F. Stesina, N. Viola, S. Corpino, M. Fioriti, Functional Analysis in Systems Engineering: methodology and applications, “Systems Engineering - Practice and Theory”, pag. 71-96, ISBN/ISSN 9789535103226, doi 10.5772/34556, RIJEKA (HRV), March 2012.
- [12] W. J. Larson, J. R. Wertz, Space mission analysis and design, Third edition, Space Technology Library, 2005.
- [13] W. J. Larson, L. K. Pranke, Human Spaceflight: Mission Analysis and Design, Space Technology Series.
- [14] S. Cresto Aleina, N. Viola, S. Ferraris, M. A. Viscio, Design Of A Reusable Space Tug, 23rd Conference of the Italian Association of Aeronautics and Astronautics (AIDAA), Politecnico di Torino, 17-19 November 2015.
- [15] S. Cresto Aleina, N. Viola, S. Ferraris, M. A. Viscio, Design Of A Reusable Space Tug, 23rd Conference of the Italian Association of Aeronautics and Astronautics (AIDAA), Politecnico di Torino, 17-19 November 2015.
- [16] VEGA User’s Manual, Issue 3/ Revision 0, March 2006.
- [17] IBM Rational DOORS,  
[www.ibm.com/software/products/it/ratidoor](http://www.ibm.com/software/products/it/ratidoor), (accessed 06.09.2016).
- [18] J-L. Voirin, Method & Tools to secure and support Collaborative Architecting of Constrained Systems, 27<sup>th</sup> International Congress of the Aeronautical Sciences, Nice, France, 2010.
- [19] Polarsys Papyrus,  
<https://www.eclipse.org/papyrus/>, (accessed 06.09.2016).
- [20] The Object Management Group, OMG Systems Modelling Language (OMG SysML), Version 1.4 2015, OMG Limited, Needham, MA (US).
- [21] S. Friedenthal, A. Moore, R. Steiner, A practical guide to SysML, The MK/OMG Press, 2014, Waltham, MA (US).
- [22] Modeliosoft, Modelio, <https://www.modelio.org>, (accessed 06.09.2016).
- [23] PTC, Integrity Modeler,  
<http://www.ptc.com/model-based-systems-engineering/integrity-modeler>, (accessed 06.09.2016).
- [24] IBM, Rational Rhapsody, <http://www-03.ibm.com/software/products/en/ratirhapfam>, (accessed 06.09.2016).
- [25] H. P. Hoffmann, Systems Engineering Best Practices with the Rational Solution for Systems and Software Engineering, version 4.1, 2011, IBM Corporation, Somers, NY, (US).
- [26] C. Bertsch, E. Ahle, E. Schumeister, The Functional Mockup Interface seen from and industrial perspective, Proc. 10<sup>th</sup> International Modelica Conference, Lund, Sweden, pp. 27-33, 2014.
- [27] E. Brusa, D. Ferretto, A. Calà, Integration of heterogeneous functional-vs-physical simulation within the industrial system design activity, Proc. 1<sup>st</sup> IEEE-ISSE Conf. Rome, Italy, pp. 303-310, 2014.
- [28] OpenModelica, <https://www.openmodelica.org>, (accessed 06.09.2016).
- [29] LMS Imagine.Lab,  
[https://www.plm.automation.siemens.com/it\\_it/products/lms/imagine-lab/amesim/](https://www.plm.automation.siemens.com/it_it/products/lms/imagine-lab/amesim/), (accessed 06.09.2016).
- [30] Mathworks, Simulink,  
<http://us.mathworks.com/products/simulink/>, (accessed 06.09.2016).